

VERIFICATION DATA GENERATING SYSTEM

Publication number: JP3151738

Publication date: 1991-06-27

Inventor: FUKUZAWA YASUKO; TAKARAGI KAZUO; SASAKI RYOICHI

Applicant: HITACHI LTD

Classification:

- **International:** H04L9/32; G06F12/00; G06F12/14; G09C1/00;
H04L9/06; H04L9/14; H04L9/32; G06F12/00;
G06F12/14; G09C1/00; H04L9/06; H04L9/14; (IPC1-7):
H04L9/06; H04L9/14

- **European:**

Application number: JP19890288887 19891108

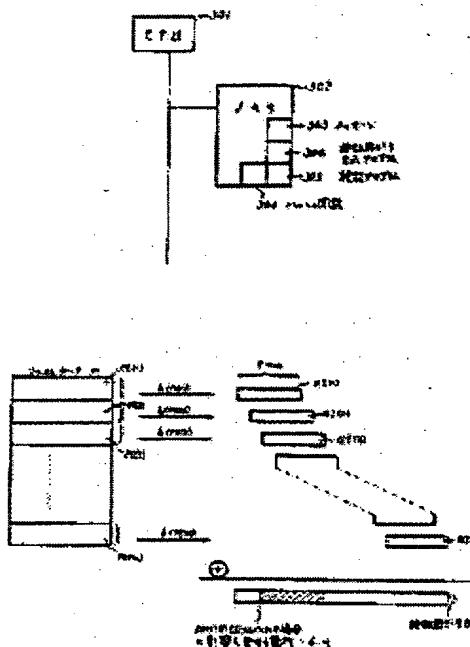
Priority number(s): JP19890288887 19891108

[Report a data error here](#)

Abstract of JP3151738

PURPOSE: To detect the presence of forgery of file content and the forged part by splitting a file data, applying logical operation while deviating one by one bit compressed sentence generated to each of split file data.

CONSTITUTION: A message 303 for an object of verification, a verification data generating program 304, a verification program 305 and a hash function 306 are stored in a memory 302 in a computer and a CPU 301 uses the data to generate and verify a verification data. A message M being an object for generating the verification data is decided into (n) as M(i) ($i=0 \dots n$), and a partial compression sentence HI(i)(p-bit) is generated with respect to the M(i) by using the hash function (h). The generated partial compression sentences HI(i) are deviated one by one bit to obtain exclusive OR and the result is used for the verification data HI in (p+n-1) bits. That is, the exclusive OR between the 2nd bit data of the HI(1) and the 1st bit data of the HI(2) is the 2nd bit data of the verification data.



Data supplied from the **esp@cenet** database - Worldwide

⑩ 日本国特許庁 (JP) ⑪ 特許出願公開
 ⑫ 公開特許公報 (A) 平3-151738

⑬ Int. Cl.⁵
 H 04 L 9/06
 9/14

識別記号 庁内整理番号

⑭ 公開 平成3年(1991)6月27日

6914-5K H 04 L 9/02 Z
 審査請求 未請求 請求項の数 6 (全12頁)

⑮ 発明の名称 検証用データ生成方式

⑯ 特 願 平1-288887
 ⑰ 出 願 平1(1989)11月8日

⑱ 発明者 福澤寧子 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内
 ⑲ 発明者 宝木和夫 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内
 ⑳ 発明者 佐々木良一 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内
 ㉑ 出願人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地
 ㉒ 代理人 弁理士 小川勝男 外1名

明細書

1. 発明の名称

検証用データ生成方式

2. 特許請求の範囲

1. 電子的なメッセージMの検証用データ生成方式において、

該メッセージMをn個に分割し、 $M = M(1) || M(2) || \dots || M(i) || \dots || M(n)$ とし、分割したn個の該メッセージ $M(i)$ ($i = 1, 2, \dots, n$)に対し、ハッシュ関数hによってpビットの圧縮文 $H(i)$ ($H(i) = h(M(i))$)($i = 1, 2, \dots, n$)を作成し、

該圧縮文 $H(i)$ ($i = 1, 2, \dots, n$)をmビット($1 \leq m \leq p$)ずつずらして論理演算を施した($p + m(n - 1)$)ビットのデータを上記メッセージMと対応する検証用データHとすることを特徴とする検証用データ作成方式。

2. 電子的なメッセージMの検証用データ生成方式において、

該メッセージMをn個に分割し、 $M = M(1)$

(1)

$|| M(2) || \dots || M(i) || \dots || M(n)$ とし、分割したn個の該メッセージ $M(i)$ ($i = 1, 2, \dots, n$)に対し、ハッシュ関数hによってpビットの圧縮文 $H(i)$ ($H(i) = h(M(i))$)($i = 1, 2, \dots, n$)を作成し、該 $H(i)$ ($i = 1, 2, \dots, n$)の左半分を $H(i)L$ ($i = 1, 2, \dots, n$)、右半分を $H(i)R$ ($i = 1, 2, \dots, n$)とし、 $H'(i)L = H(j)L$ ($1 \leq j \leq n$ であり、 $H'(j)L \neq H'(p)L$ ($p < i$))となるように $H(i)L$ を再配置した $H'(i)L$ ($i = 1, 2, \dots, n$)を生成し、 $H'(i)R = H(j)R$ は、 $1 \leq j \leq n$ であり、 $H'(j)R \neq H'(p)R$ ($p < i$)であり、 $H'(f)R$ ($i - 2 \leq f < i, i < f \leq i + 2$) $\neq H(k)R$ ($j - 2 \leq k < j, j < k \leq j + 2$)であり、 $H'(m)R$ ($j - 5 \leq m \leq j - 1$) $\neq H(n)R$ ($i + 1 \leq n \leq i + 5$)となるように $H(i)R$ を再配置した $H'(i)R$ ($i = 1, 2, \dots, n$)を生

(2)

成し、

該再配列した結果を $H'(i)$ ($i = 1, 2 \dots n$) とし、

該 $H'(i)$ ($i = 1, 2 \dots n$) を m ビットずつずらして論理演算を施した ($p + m(n - 1)$) ビットのデータを上記メッセージ M と対応する検証用データ H' とすることを特徴とする検証用データ作成方式。

3. 請求項 2 に記載の検証用データ生成方式において、

作成した p ビットの部位圧縮文 $H(i)$ ($i = 1, 2 \dots n$) の各々を、

s (p の公約数であり、 $s \neq 1, 2, p$) 個で分割し、 $H(i)(1), H(i)(2), \dots, H(i)(r), \dots, H(i)(s)$ ($i = 1, 2 \dots n$) とし、

$H'(i)(1) = H(j)(1)$ ($1 \leq j \leq n$ であり、 $H'(j)(1) \neq H'(p)(1)$ ($p < i$)) となるように $H(i)(1)$ を再配列した $H'(i)(1)$ ($i = 1, 2 \dots n$) を生成し、

(3)

ータとすることを特徴とする電子捺印方式。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、電子化されたファイルの検証用データ生成方式に関する。

〔従来の技術〕

電子データの活用や保存が盛んになるに従い、ファイルの正当性を認証する技術はますます重要になる。ファイルの内容を圧縮文によって表す技術は、ファイル内容を確認する有効な技術の一つである。これは大量のファイルデータをハッシュ関数によって圧縮し、64ビット程度の認証子（圧縮文）を作成する技術である。ファイルデータの内容が1ビットでも変わると、全く異なった圧縮文を生成してしまうため、データ改ざんの有無を検知することができる。この技術に、公開鍵暗号を応用し、上記記載の電子取引の安全性を確保する電子捺印が提案されている。

〔発明が解決しようとする課題〕

しかし、改ざんの有無を検知することができて

(5)

$1 < r \leq s$ の r について、

($1 \leq x \leq n$) であり、 $H'(i)(r)$ が影響を及ぼす p ビットの範囲に存在する x に関し、

$H'(i)(r) \neq H(x)(r)$ ないように $H(i)(r)$ を再配列した $H(i)'(r)$ ($i = 1, 2 \dots n$) を生成し、

該再配列した結果を $H'(i)$ ($i = 1, 2 \dots n$) とする再配列方式。

4. 請求項 2 もしくは 3 に記載の検証用データ生成方式において、

n 個に分割したメッセージ

$n > (2p/s - 1)^2$ の関係式が成立する検証用データ生成方式。

5. ファイル M の圧縮文に、請求項 1 乃至 4 のいずれかに記載の、検証用データを付加することを特徴とするファイル認証方式。

6. 請求項 1 乃至 5 のいずれかに記載のファイルの圧縮文に、時刻等の状況データを付加したデータを、公開鍵暗号の秘密鍵を用いて、公開鍵暗号で暗号処理し、これを該ファイルの認証デ

(4)

も、改ざんの箇所を検知することはできない。

この問題に対処するため、改良案を考案していた（特願昭62-321220号）。これはファイルを階層化し、階層化した個々のファイルデータに対して圧縮文を生成、保存することで、後日の改ざんを検知する。しかし、この方法だと、個々の圧縮文を保存するため、保存すべき情報量が多くなるという不満足な点があつた。

〔課題を解決するための手段〕

上記問題に対し、検証用データ生成方式を考案した。これは、ファイルデータを分割し、分割した個々のファイルデータに対して作成した圧縮文を1ビットずつずらして論理演算を行う。あるいは、検知確率を向上させるために、個々の圧縮文を2つに分割し、最適に配列し、配列し直した各圧縮文に対し、論理演算を行う。

〔作用〕

前記技術的手段により、次の効果が生じる。

1. ファイルの検証用データ（e.g. 416ビット）生成後、ファイルデータが改ざんされた場

(6)

合、 $\left(1 - \frac{1}{2^{416}}\right)$ の確率で改ざんの有無を検知することができる。

2. ファイル改ざん前後の検証用データの相違によって、ファイルデータの改ざん位置をかなりの確率で検知することが可能になる。

〔実施例〕

第1図～第9図において、本発明の実施例を示す。

〔実施例1〕

第1図～第4図において、電子的なメッセージMの検証用データ生成方式、およびメッセージ改ざん検証の一例を示す。

第1図は、検証用データ生成の一方法を示すフロー図である。第2図は、メッセージの改ざんを検知する一方法を示すフロー図である。

第3図は、処理を行う計算機の一例である。

第4図は、第1図の検証用データ生成の実際のイメージを示す。

(7)

step 106 : i に 1 を加え、step 102 に進む。
step 107 : カウント i を 0 に設定する。

step 108 : カウント i が、 $i < n$ ならば step 109 に進み、 $i \geq n$ ならば step 111 に進む。

step 109 : 作成した部位圧縮文 H I (i) を 1 ビットずつずらせて排他的論理和を求め、これを $(p + n - 1)$ ビットの検証用データ H I' とする。つまり、H I (1) の 2 ビット目と H I (2) の 1 ビット目の排他的論理和が検証用データの 2 ビット目となる。

H I (1) の 3 ビット目と H I (2) の 2 ビット目と。

H I (3) の 1 ビット目の排他的論理和が検証用データの 3 ビット目となる。

step 110 : i に 1 を加え、step 108 に進む。
step 111 : 検証用データ H I' を出力する。
step 112 : 終わり。

次に、上記の検証用データ H I' 作成時のメッセージ M と現時点でのメッセージ M' が同等であるかを検証する例を第2図のフローに従つて示す。

(9)

第3図において、計算機上のメモリ 302 に、検証対象のメッセージ 303、検証用データ生成プログラム 304、検証プログラム 305、およびハッシュ関数 306 が蓄えられており、これらを用いて CPU301 によって検証用データ生成と検証を行う。検証用データ生成手段を第1図のフローのステップ (step) に従つて示す。

step 100 : 始め

step 101 : 検証用データ生成の対象となるメッセージの名称 M を設定し、メッセージを n 個に分割し、個々を M(i) ($i = 0 \dots n$) とする。また、カウント i を 0 に設定する。

step 102 : カウント i が、 $i < n$ ならば step 103 に進み、 $i \geq n$ ならば step 107 に進む。

step 103 : メッセージ M(i) を読み込む。

step 104 : M(i) に対して、部位圧縮文 H I (i) (p ビット) をハッシュ関数 h を用いて生成する。

step 105 : 部位圧縮文 H I (i) をメモリ 302 上に退避する。

(8)

step 200 : 始め

step 201 : 既に生成済みの M の検証用データ H I' を入力する。

step 202 : 検証の対象であるメッセージ M' について、step 100 から step 112 に従い、検証用データを生成し、これを H I'' とする。

step 203 : 検証用データ H I' と H I'' を比較し、不一致部分を検知する。一致した場合は step 204 に進み、不一致の場合には step 205 に進む。

step 204 : メッセージ M と M' は同一であると判定し、step 207 に進む。

step 205 : メッセージ M と M' は同一でないと判定する。

step 206 : 検知した不一致部分位置から、メッセージ M' の改ざん部位を検知する。

例えば、第4図において、H I と H I'' を比較すると d_H の位置が影響を受けていた場合、H I (3) と H I'' (3) が一致しなかつたことが自明であり、この結果 M (3) が改ざんされたことが

(10)

わかる。

step 207：終わり。

[実施例 2]

第5図～第8図において、電子的なメッセージMの検証用データ生成方式、およびデータ改ざんを検証する他の例を示す。

第5図は、検証用データ生成の一方法を示すフロー図である。第6図は、メッセージの改ざんを検知する一方法を示すフロー図である。

第7図、第8図は、検証用データ生成の実際のイメージを示す。

第5図、および第7図、第8図において、検証用データ生成の手順を示す。

メッセージMをn個に分割し、各分割メッセージに対してpビットの部分圧縮文を生成し、部分圧縮文をs個に分割し、これを再配置して検証用データの生成を行う。この時、再配置における分散を高めるために、例えばn, s, pは次の関係式が成り立つようとする。

$$n > (2p/s - 1)^2$$

(11)

step 508：カウントiが、 $i < 26$ ならばstep 509に進み、 $i \geq 26$ ならばstep 512に進む。

step 509：作成した部位圧縮文H II(i) ($i = 1, 2 \dots 26$) の左側3ビットをH II(i)L、右側3ビットをH II(i)Rとする。

$H II'(i)L = H II(i)L$ ($i = 1, 2 \dots 26$) とする。

$H II'(i)R = H II(j)R$ ($i = 1, 2 \dots 26$) とし、jを次のルールに従い再配置する。

(1) $1 \leq j \leq 26$ であり、

(2) $H II(j)R \neq H II'(p)R$ ($p < j$) であり、

(3) $H II'(i)R$ が影響を与える $H II'(k)R$ ($i - 2 \leq k < i, i < k \leq i + 2$) は、

$H II'(j)L$ が影響を与える $H II(f)R$ ($j - 2 \leq f < j, j < f \leq j + 2$) でなく、

(4) $H II(j)L$ が影響を与える $H II'(m)R$ ($j - 5 \leq m \leq j - 1$) には、 $H II'(i)L$ が影響を与える $H II(n)R$ ($i + 1 \leq n \leq i + 4$) ではない。

(13)

ここでは、ファイルMを26個に分割し、生成する各部位圧縮文は6ビットとし、各部位圧縮文は2つに分割して再配置する。各部位圧縮文より生成する検証用データ31ビットとする。

step 500：始め

step 501：検証用データ生成の対象となるメッセージの各称Mを設定し、メッセージを26個に分割し、個々をM(i) ($i = 1, 2 \dots 26$) とする。また、カウントiを0に設定する。

step 502：カウントiが、 $i < 26$ ならばstep 503に進み、 $i \geq 26$ ならばstep 507に進む。

step 503：M(i)を読み込む。

step 504：M(i)に対して、ハッシュ関数hを用いて部位圧縮文H II(i) (6ビット)を作成する。

step 505：部位圧縮文H II(i)をメモリ302上に退避する。

step 506：iに1を加え、step 501に進む。

step 507：カウントiを0に設定する。

(12)

step 510：作成した部位圧縮文H II(i)を1ビットずつずらせて排他的論理和を求め、これを $(p + n - 1)$ ビットの検証用データH IIとする。つまり、H II'(1)の2ビット目とH II'(2)の1ビット目の排他的論理和が検証用データの2ビット目となる。H II'(1)の3ビット目とH II'(2)の2ビット目と、H II'(3)の1ビット目の排他的論理和が検証用データの3ビット目となる。

step 511：iに1を加え、step 508に進む。

step 512：検証用データH IIを出力する。

step 513：終わり。

上記手順に従い生成した検証用データの例が第7図である。

次に、上記の検証用データH II作成時のメッセージMと現時点でのメッセージM'が同等であるかを検証する例を第6図のフローに従つて示す。

step 600：始め

step 601：既に生成済みのMの検証用データH IIを入力する。

(14)

step 602：検証の対象であるメッセージ M'' について、step 500 から step 513 に従つて、検証用データ H'' 生成と同じ型の再配列を行い、メッセージ M'' の検証用データを生成し、これを H''' とする。

step 603：検証用データ H'' と H''' を比較し、一致した場合は step 604 に進み、不一致の場合には step 605 に進む。

step 604：メッセージ M と M'' は同一であると判定し、step 607 に進む。

step 605：メッセージ M と M'' は同一でないと判定される。

step 606：ファイルデータ改ざん前後の改ざん検知用圧縮文 H と H''' の比較する。 $M(5)$ が改ざんされた場合には、検証用データ H''' において、 D_1 、および D_2 の部分で一致しない。

従つて、改ざん部位の構成より、次のように判断できる。

改ざん部位 = $D_1 \cap D_2$

= $(H(3)LUH(4)LUH(5)LUH(6)LUH(7))L$
(15)

持する検証用データが多くなり、一方、改ざん位置の検知確率は向上する)。

[変形例 3]

実施例 2において、分割した部位の各圧縮文を、3以上に複数に分割する。例えば、 H'' を 3 分割し $H'' L(i)$, $H'' M(i)$, $H'' R(i)$ ($i = 1, 2 \dots n$) とし、

$H''(i)L = H''(i)L$ ($i = 1, 2 \dots 26$) とする。

$H''(i)M = H''(j)R$ ($i = 1, 2 \dots 26$) とし、 j を次のルールに従い再配置する。

(1) $1 \leq j \leq 26$ であり、

(2) $1 \leq k \leq 26$ であり、 $H''(j)M$ が影響を与える範囲に存在する k に関して、 $H''(j)M \neq H''(k)M$ とする。

また、 $H''(i)R = H''(j)R$ ($i = 1, 2 \dots 26$) とし、 j を次のルールに従い再配置する。

(1) $1 \leq j \leq 26$ であり、

(2) $1 \leq k \leq 26$ であり、 $H''(j)R$ が影響を

(17)

$U(H(7)RUH(10)RUH(13)RUH(16)R)$

□

$(H(17)LUH(18)LUH(19)LUH(20)LUH(21))LU(H(23)RUH(26)RUH(5)RUH(1)RUH(11)R)$

= $H(5)$

$M(5)$ が改ざんされたことが検知できる。

ただし、ここでの \cap は、論理積であり、 L と R が対となつていることを意味する

step 607：終り。

[変形例 1]

実施例 1、実施例 2 の検証用データ生成において、生成した各部位圧縮文を、排他的論理和以外の論理演算（論理和、論理積等）によって処理しても同等の機能を実現することができる。

[変形例 2]

実施例 1、実施例 2 の検証用データ生成において、生成した各部位の圧縮文を m ($1 \leq m \leq p$) ビットずつずらせて論理演算処理を行つても同等の機能を実現することができる (m が多いほど保

(16)

与える範囲に存在する k に関して、 $H''(j)R \neq H''(k)R$ とする。

再配置した部位圧縮文 $H''(i)$ を 1 ビットずつずらせて排他的論理和を求め、これを $(p+n-1)$ ビットの検証用データ H'' とするこも可能である。

[変形例 4]

実施例 1、実施例 2 で生成の検証用データ生成方式は、電子取引認証における電子捺印に利用することができる。

step 911：取引伝票 900 を 353 の部位に分割し、各部位の圧縮文 (64 ビット) を作成し、改ざん部位検知用圧縮文 903 (416 ビット) を、実施例 1、あるいは 2 によって作成する。

step 912：取引伝票 900 の圧縮文 902 ($h(M)$) を作成する。

step 913：(圧縮文 902 || 改ざん部位検知用圧縮文 903 || 32 ビットの時刻等の状況データ 904) を電子捺印文 (512 ビット) 901 とし、公開鍵暗号で暗号処理する。

(18)

〔変形例 5〕

実施例 1、実施例 2 で生成の検証用データ生成方式は、ファイル認証における認証子として利用することができる。

〔変形例 6〕

検証用データの生成、および検証を IC カード上で実施することも可能である。

〔変形例 7〕

生成した検証用データを、IC カードに保存することも可能である。

〔変形例 8〕

実施例 2において、検証用データを用いて検証を行う場合に、確率的評価を加えることが可能である。実施例 2 では、メッセージ M(5) の改ざんに伴い、D1、および D2 に影響が生じているが、

D1 に最も影響を与える確率が高いのは HLL(5)、HRL(10) であり。

D2 に最も影響を与える確率が高いのは HLL(19)、HRL(5) であることから、

$$\text{改ざん部位} = D1 \cap D2$$

(19)

確率で検知することが可能になる。

4. 図面の簡単な説明

各図は本発明の実施例を示し、第 1 図は検証用データ生成の一方法を示すフロー図、第 2 図は、メッセージの改ざんを検知する方法を示すフロー図、第 3 図は、処理を行う計算機の一例を示すブロツク図、第 4 図は、第 1 図の検証用データ生成の実際のイメージを示す説明図、第 5 図は、検証用データ生成の他の方法を示すフロー図、第 6 図は、メッセージの改ざんを検知する一方法を示すフロー図、第 7 図と第 8 図は、第 5 図の検証用データ生成の実際のイメージを示す説明図、第 9 図は、検証用データ生成方式を電子捺印に適用した実施例の説明図である。

代理人 弁理士 小川勝男



(21)

$$= (HLL(5) \cup HRL(10))R$$

□

$$(HLL(19) \cup HRL(5))R$$

$$= H(5)$$

と検証することができる。

複数箇所の改ざん場所検知等の適用に有効である。

〔効果〕

本発明において、ファイル分割情報があり、かつファイル改ざん前後のファイル圧縮文、および改ざん検知用圧縮文が生成できる場合、次のような効果が得られる。

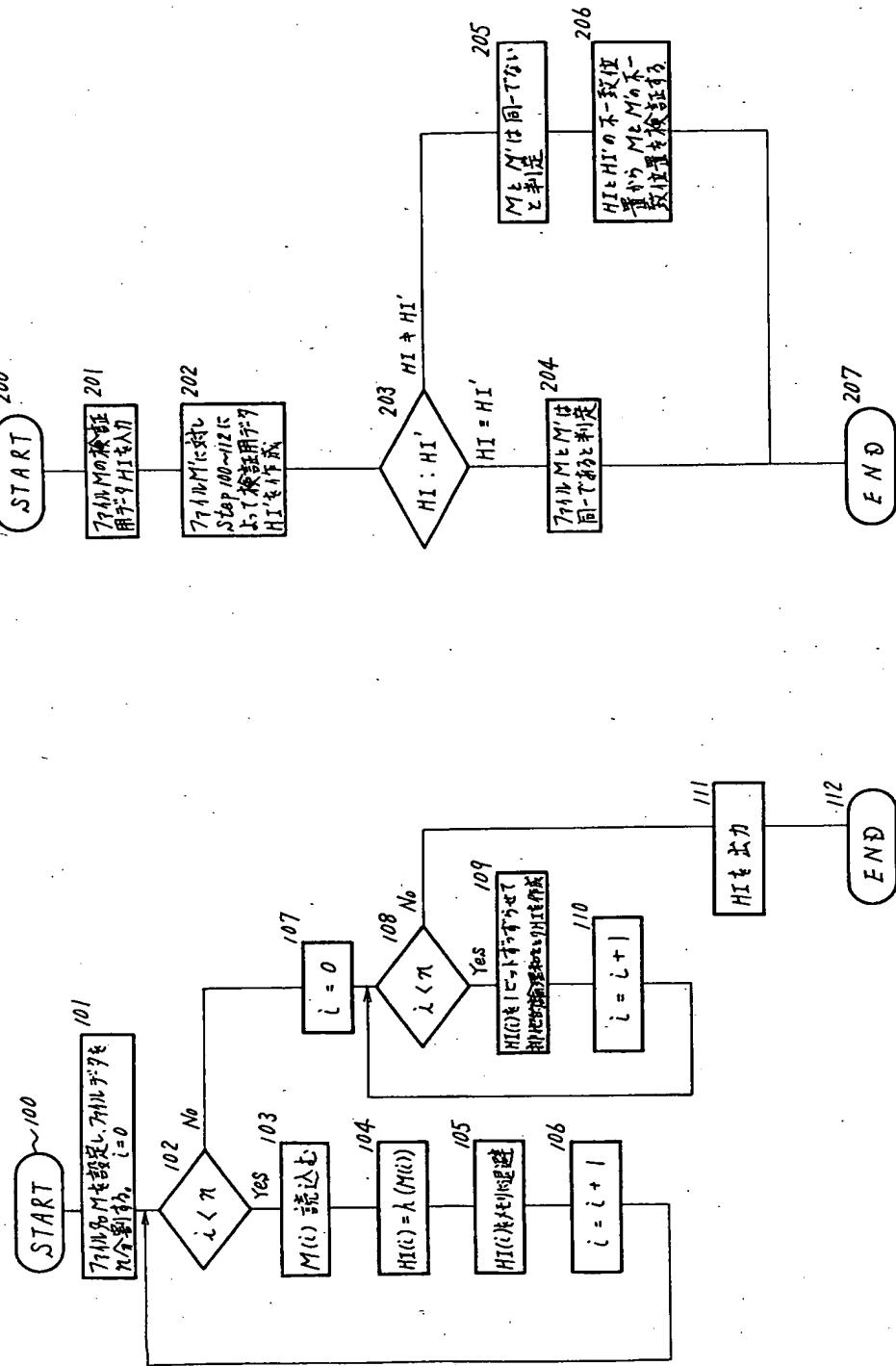
1. 改ざん前のファイルの改ざん検知用圧縮文 (ex. 416 ビット) 生成後、ファイルデータが改ざんされた場合、

$$\left(1 - \frac{1}{2^{416}}\right) \text{ の確率で改ざんの有無を検知する}$$

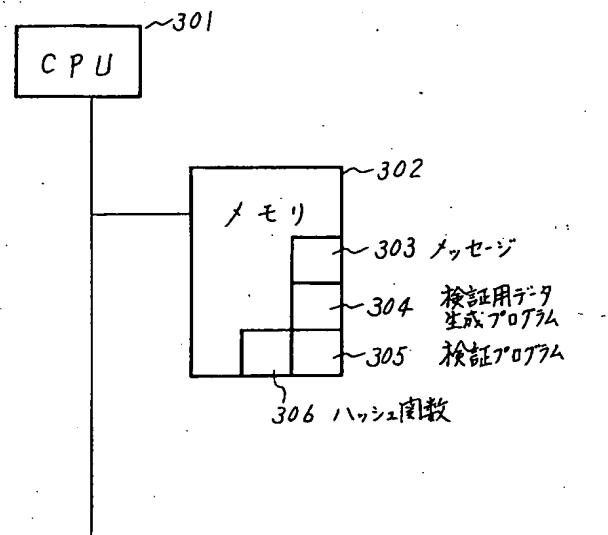
2. ファイル改ざん前後の改ざん検証用圧縮文により、ファイルデータの改ざん位置をかなりの

(20)

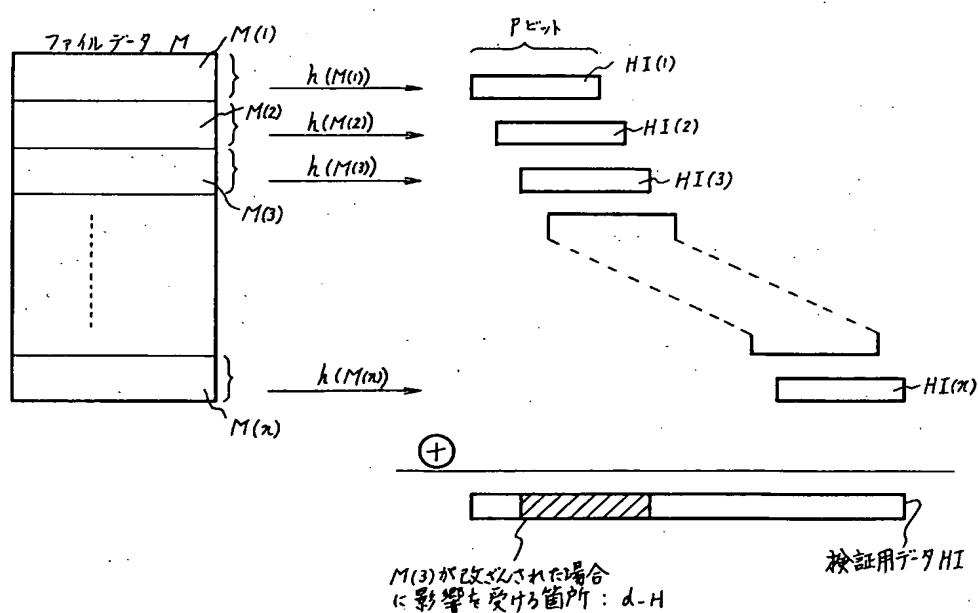
第1図 第2回



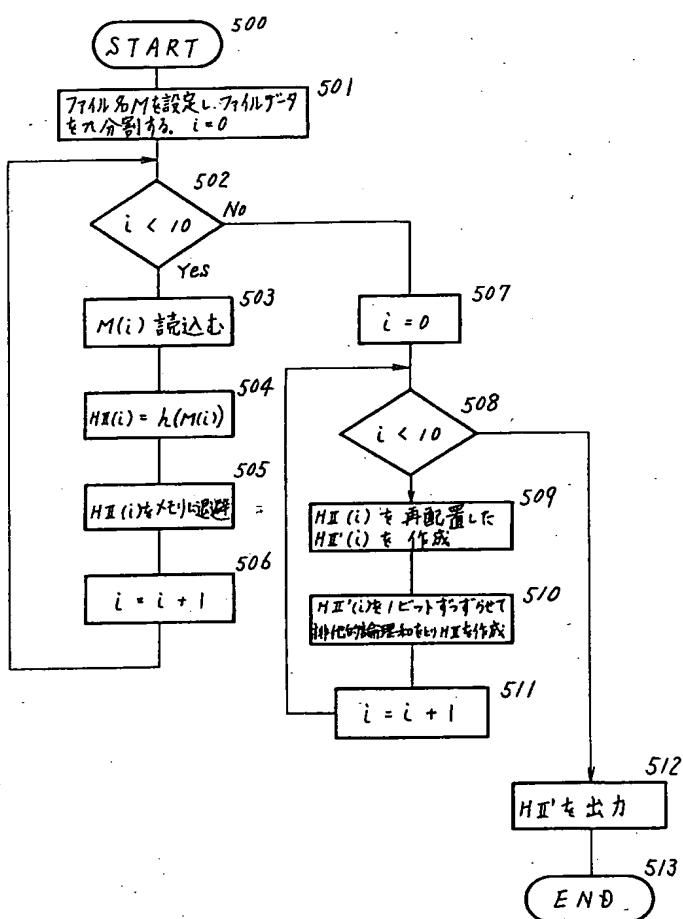
第 3 図



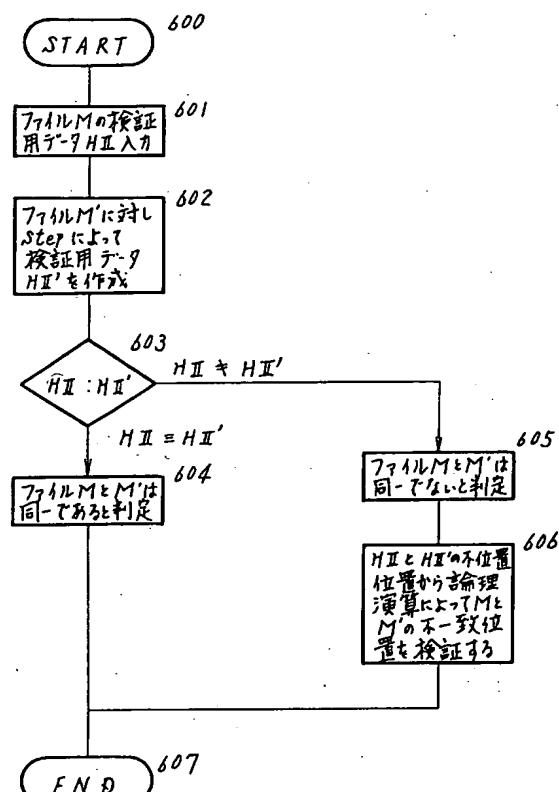
第 4 図



第 5 図

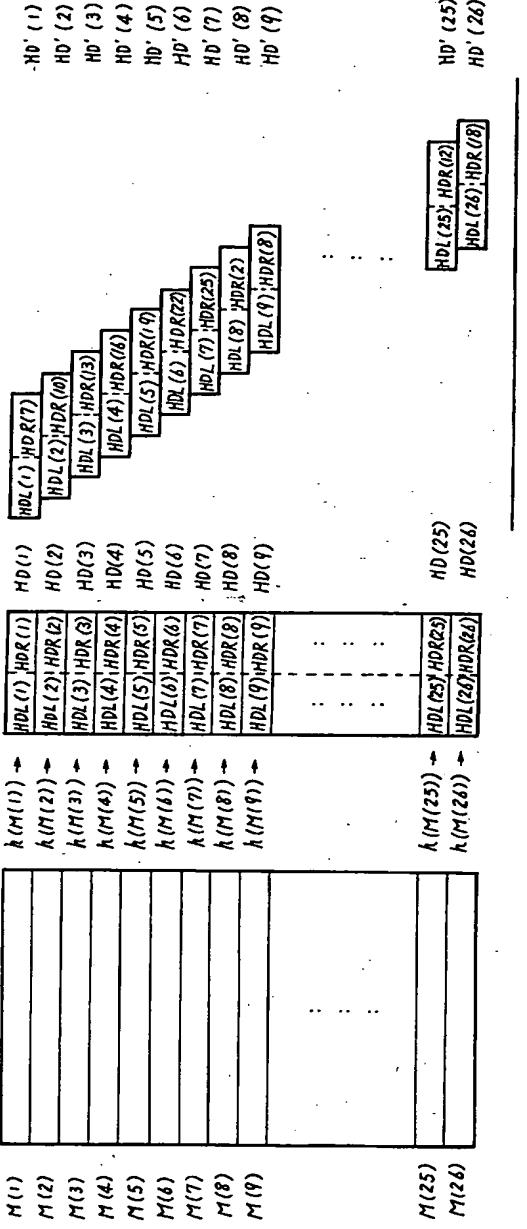


第 6 図



第 7 図

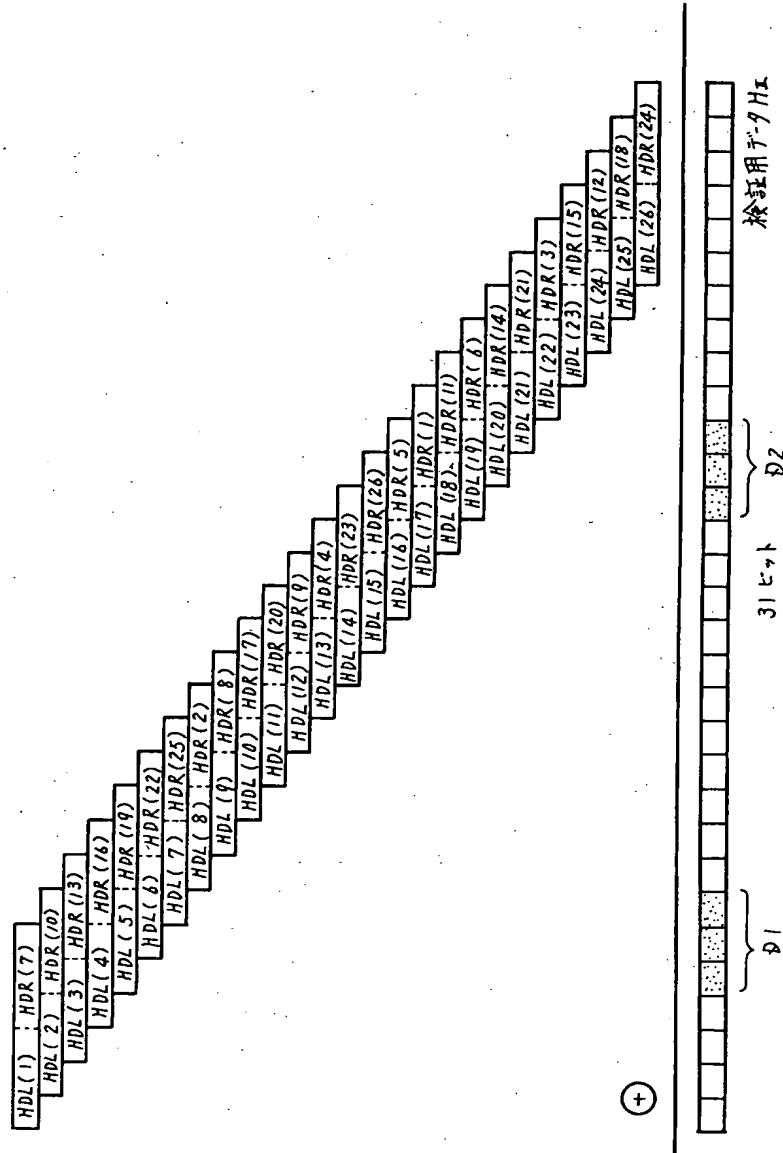
7-1 NFT-9 M



※知用テ-ア HD



四



第 9 図

